

KOMPRESI CITRA MENGGUNAKAN METODE FRAKTAL



Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada Jurusan
Informatika Fakultas Komunikasi dan Informatika

Oleh:

MUCHAMMAD ARIFIN
L200130053

PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2018

HALAMAN PERSETUJUAN
KOMPRESI CITRA MENGGUNAKAN METODE FRAKTAL

PUBLIKASI ILMIAH

oleh:

MUCHAMMAD ARIFIN
L 200 130 053

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing



Diah Priyawati, M.Eng.

HALAMAN PENGESAHAN
KOMPRESI CITRA MENGGUNAKAN METODE FRAKTAL

OLEH
MUCHAMMAD ARIFIN
L 200 130 053

Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Selasa, 03 April 2018
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. Diah Priyawati, M.Eng.
(Ketua Dewan Penguji)
2. Dr. Ir. Bana Handaga, M.T.
(Anggota I Dewan Penguji)
3. Hernawan Sulistyanto, S.T, M.T.
(Anggota II Dewan Penguji)

(.....)
(.....)
(.....)

Publikasi ilmiah ini telah diterima sebagai salah satu persyaratan
Untuk memperoleh gelar sarjana
Tanggal 03 April 2018
Mengetahui,



Dekan
Fakultas Komunikasi dan Informatika

Nurgiyatna, S.T., M.Sc., Ph.D.
NIK : 881



Ketua Program Studi
Informatika

Dr. Heru Supriyo, M.Sc.
NIK: 970

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam naskah publikasi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 03 April 2018

Penulis



MUCHAMMAD ARIFIN

L 200 130 053



UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

SURAT KETERANGAN LULUS PLAGIASI

No Surat .188/A-3-13/inf-FKI/V/2018

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : Muchammad Arifin
NIM : L200130053
Judul : KOMPRESI CITRA MENGGUNAKAN METODE FRAKTAL
Program Studi : Informatika
Status : Lulus

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 10 Mei 2018

Biro Skripsi Informatika




Ihsan Cahyo Utomo, S.Kom., M.Kom.



UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

Amor <https://ums.ac.id> | [Feedback Studio](#) | KOMPRESI CITRA MENGGUNAKAN METODE FRAKTAL

KOMPRESI CITRA MENGGUNAKAN METODE FRAKTAL

Abstrak

Pada saat ini kebutuhan akan teknologi informasi begitu pesat. Adanya kompresi citra sangat dibutuhkan dalam dunia foto digital. Penelitian ini bertujuan untuk melakukan kompresi citra dengan metode fraktal guna mengurangi kapasitas penyimpanan dengan kualitas citra yang baik. Kompresi fraktal adalah kompresi yang memiliki kemiripan dari citra itu sendiri secara keseluruhan. Metode *quadtree decomposition* digunakan untuk menentukan ukuran blok bervariasi sesuai dengan fitur gambar dan dipisahkan menjadi beberapa blok berbeda. *Self-similarity* dilakukan untuk mencari kemiripan dari blok secara keseluruhan. Setelah kompresi, *huffman coding* diterapkan pada citra yang terkompresi sehingga kinerjanya akan lebih baik. Dalam hal ini ukuran kuantitas yang berbeda ditemukan dengan melewatkan citra dengan dimensi dan format yang berbeda, dan format citranya baik untuk kompresi. Maka penelitian ini menggunakan penghitungan MSE dan PSNR untuk mengetahui keberhasilan kompresi fraktal dengan metode kompresi lain. Penelitian ini juga dibandingkan dengan metode kompresi DCT. Metode kompresi fraktal menghasilkan nilai yang lebih baik yaitu nilai untuk rata-rata MSE adalah 11,07 dB dan nilai rata-rata PSNR adalah 38,48 dB dengan nilai rata-rata rasio kompresinya adalah 95,05%.

Kata Kunci: Fraktal, huffman, MSE, PSNR, *quadtree decomposition*, rasio.

Abstract

Page 1 of 18 | Word Count: 3291

Match Overview

15%

Rank	Source	Score
1	repositary.ums.ac.id	4%
2	repositary.ums.ac.id	3%
3	whatsapp.blogspot.com	1%
4	etheses.ums.ac.id	1%
5	journal.portuguesia.org	1%
6	id.eprints.ums.ac.id	1%
7	www.dicthoc.com	1%
8	gipsos.com	<1%
9	Submitted to Universitas	<1%
10	matlabberita.blogspot	<1%
11	com.fanqidi.ac.id	<1%
12	www.bruceinc.org	<1%

Each only Report | High Resolution

KOMPRESI CITRA MENGGUNAKAN METODE FRAKTAL

Abstrak

Pada saat ini kebutuhan akan teknologi informasi begitu pesat. Adanya kompresi citra sangat dibutuhkan dalam dunia foto digital. Penelitian ini bertujuan untuk melakukan kompresi citra dengan metode fraktal guna mengurangi kapasitas penyimpanan dengan kualitas citra yang baik. Kompresi fraktal adalah kompresi yang memiliki kemiripan dari citra itu sendiri secara keseluruhan. Metode *quadtree decomposisi* digunakan untuk menentukan ukuran blok bervariasi sesuai dengan fitur gambar dan dipisahkan menjadi beberapa blok berbeda. *Self-similarity* dilakukan untuk mencari kemiripan dari blok secara keseluruhan. Setelah kompresi, *huffman coding* diterapkan pada citra yang terkompresi sehingga kinerjanya akan lebih baik. Dalam hal ini ukuran kuantitas yang berbeda ditemukan dengan melewati citra dengan dimensi dan format yang berbeda, dan format citranya baik untuk kompresi. Maka penelitian ini menggunakan penghitungan MSE dan PSNR untuk mengetahui keberhasilan kompresi fraktal dengan metode kompresi lain. Penelitian ini juga dibandingkan dengan metode kompresi DCT. Metode kompresi fraktal menghasilkan nilai yang lebih baik yaitu nilai untuk rata-rata MSE adalah 11,07 dB dan nilai rata-rata PSNR adalah 38,48 dB dengan nilai rata-rata rasio kompresinya adalah 95.05%.

Kata Kunci: Fraktal, huffman, MSE, PSNR, *quadtree decomposisi*, rasio.

Abstract

At this time the need for information technology so rapidly. The existence of image compression is needed in the world of digital photos. This study aims to perform image compression with fractal method to reduce storage capacity with good image quality. Fractal compression is a compression that resembles the image itself as a whole. The method of quadtree decomposition is used to determine the block size varies according to the image feature and is separated into several different blocks. Self-similarity is done to find the resemblance of the whole block. After compression, huffman coding is applied to the compressed image so that its performance will be better. In this case different quantity sizes are found by skipping images with different dimensions and formats, and the image format is good for compression. So this study uses the calculation of MSE and PSNR to determine the success of fractal compression with other compression methods. This study was also compared with the DCT compression method. Fractal compression method yields better value that is the value for the mean of MSE is 11,07 dB and the mean value of PSNR is 38,48 dB with the average value of compression ratio is 95.05%.

Keywords: Fractal, huffman, MSE, PSNR, quadtree decomposition, ratio.

1. PENDAHULUAN

Pada saat ini kebutuhan akan teknologi informasi berkembang begitu pesat. Informasi itu sendiri terdiri atas beberapa bentuk antara lain adalah citra, audio dan video. Pertukaran atau komunikasi data pada saat ini sangatlah penting dan sering dilakukan oleh semua orang terutama untuk bertukar informasi tentang pemandangan alam seperti gunung, laut, pantai, dan lain-lain. Untuk mendapatkan gambaran sebuah pemandangan maka dibutuhkan sebuah kamera untuk merekam

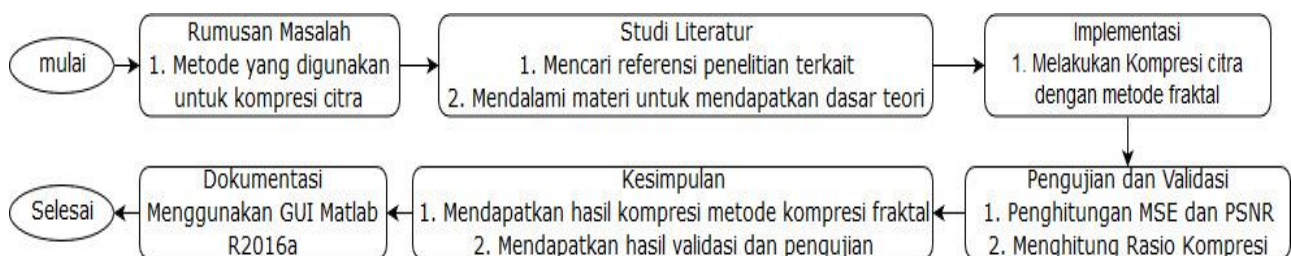
kejadian yang dibutuhkan untuk mendapatkan informasi yang diperlukan. Informasi tersebut adalah berupa sebuah citra/foto. Terkadang citra tersebut memiliki ukuran yang relatif besar untuk tersimpan di memori yang terbatas. Maka dari itu proses kompresi citra dibutuhkan untuk dapat mengurangi ukuran citra agar dapat tersimpan pada memori yang terbatas dengan kualitas citra yang baik dan mendapatkan rasio kompresi yang baik.

Pada penelitian sebelumnya didapatkan kompresi citra dengan kompresi DCT (*Discrete Cosine Transform*) didapatkan hasil kualitas citra dengan nilai MSE sebesar 77,59 dB dan PSNR sebesar 25, 32 dB (Mahmudi, Adnan, 2017). Penelitian kompresi citra dengan algoritma kompresi fraktal dengan *sequential* dan *parallel* untuk kompresi citra didapatkan rata-rata rasio kompresi sebesar 41,79 % (Hiryanto, 2010). Dalam penelitian yang dilakukan (Harjoko, 2006) menggunakan kompresi citra fraktal mendapatkan hasil ukuran yang relatif kecil tetapi kualitas citra mengalami penurunan yang signifikan.

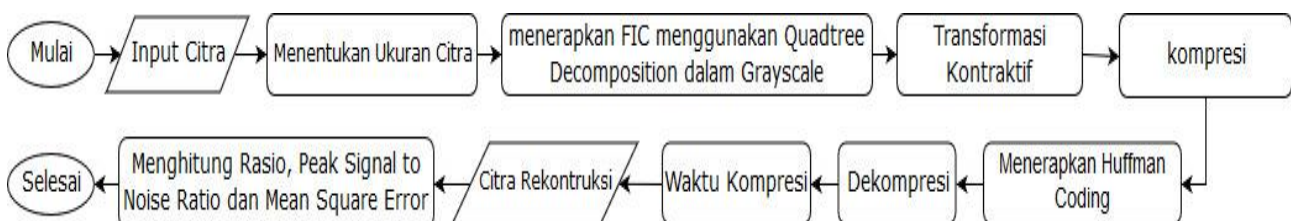
Dalam penelitian ini akan digunakan kompresi citra fraktal dengan menggunakan *quadtree decomposition* dan *huffman coding* diharapkan mampu mendapatkan ukuran citra relatif kecil dengan kualitas citra hasil kompresi yang bagus dan rasio kompresi yang baik. Validasi penghitungan MSE dan PSNR adalah parameter untuk mengetahui berapa besar kualitas citra yang didapatkan setelah melewati proses kompresi.

2. METODE

Berdasarkan masalah yang terjadi seperti kualitas citra, ukuran citra dan rasio kompresi dari berbagai kompresi yang telah dilakukan, maka penelitian ini diharapkan dapat mengatasi permasalahan tersebut dengan metode yang akan diterapkan. Penelitian ini mengikuti beberapa tahap seperti yang ditunjukkan pada Gambar 1 merupakan tahapan langkah penelitian. Gambar 2 merupakan tahapan langkah sistem yang telah dibuat.



Gambar 1. Langkah Penelitian



Gambar 2. Langkah Sistem

2.1 Input Citra digital

Input atau memasukkan citra digital. Citra digital dapat dinyatakan sebagai suatu fungsi dua dimensi $f(x,y)$, dengan x maupun y adalah posisi koordinat sedangkan f merupakan amplitudo pada posisi (x,y) yang sering dikenal sebagai intensitas atau grayscale. Resolusi pixel merupakan penghitungan jumlah piksel dalam sebuah citra digital. Dalam sebuah citra dengan tinggi N piksel dan lebar M piksel berarti memiliki resolusi sebesar $N \times M$ dengan kata lain citra digital $N \times M$ mempunyai NM buah pixel (Mukharrom Edisuryana, 2013). Citra keabuan (*grayscale*) merupakan citra digital yang hanya memiliki satu nilai kanal pada pixelnya, dengan kata lain nilai bagian $r = g = b$. Nilai tersebut digunakan untuk menunjukkan intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih (Putra, 2010). Banyaknya warna tergantung pada jumlah bit yang disediakan dimemori untuk menampung kebutuhan warna ini (T.Sutoyo, 2009). Jumlah warna pada citra keabuan adalah 256, karena citra keabuan jumlah bitnya adalah 8, sehingga jumlah warnanya adalah $2^8 = 256$, nilainya berada pada jangkauan 0 – 255 (Mukharrom Edisuryana, 2013). Berikut adalah contoh citra keabuan (*grayscale*), seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Citra keabuan (*grayscale*)

2.2 Mengambil Ukuran Citra

Kemudian dilakukan pencarian ukuran citra, ukuran citra untuk kompresi fraktal biasanya membagi beberapa blok yang disebut blok *range* yang berukuran 4×4 atau 8×8 agar tidak saling beririsan dan mempermudah membentuk persegi. Selanjutnya dibuat blok domain yang ukurannya 2 kali dari blok *range*. Blok *range* dan blok domain diambil tidak saling beririsan agar waktu pencocokan lebih cepat. Karena jika semua beririsan waktu pencocokan yang didapatkan akan lebih lama (Harjoko, 2006). Untuk pencarian blok *range* yang mempunyai kemiripan tinggi dengan blok domain dapat diukur tingkatnya dengan menggunakan persamaan (1):

$$d_{rms} = \frac{1}{n} \sqrt{\sum_{i=1}^n (r_i - d_i)^2} \dots\dots\dots(1)$$

Dimana :

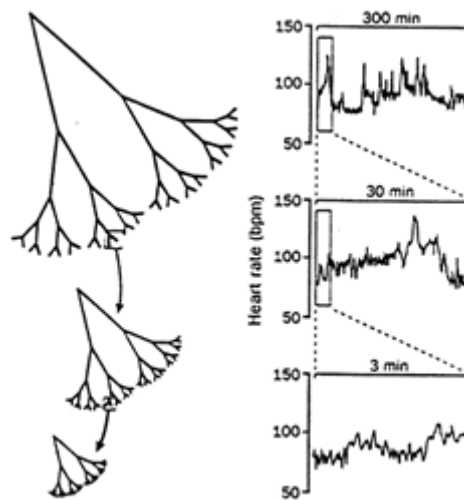
r_i = nilai intentitas *range*

d_i = nilai intentitas domain

n = jumlah *pixel* pada citra

2.3 Fraktal

Fraktal berarti adalah pecah (*broken*) atau tidak teratur (*irregular*). Fraktal pertama kali dikenalkan oleh Benoit B Mandelbrot (1973). Fractal adalah alat yang tepat untuk mempresentasikan permodelan fenomena alam (*natural phenomenon*), objek alam (Pandey dan Seth, 2014; Putra, 2010). Objek fraktal memiliki kemiripan dengan dirinya sendiri dalam skala yang berbeda, artinya bagian yang lebih kecil pada objek tersebut akan mirip dengan objek itu sendiri bila dilihat secara keseluruhan. Gambar 5. Bila diperhatikan gambar diperoleh dari penerapan aturan sama yang terulang (proses rekursif) dan perulangan ini berlangsung tanpa batas dengan berbagai ukuran/skala



Gambar 4. *Self similarity* fraktal

Bagian penting dari fraktal adalah *self similarity* yaitu berapapun skala/ukuran yang digunakan untuk menghasilkan objek fraktal maka objek yang dihasilkan akan tetap memiliki keseluruhan objek fraktal (Putra, 2010). Hasil iterasi akan selalu lebih kecil daripada sebelumnya dan akan diperoleh hasil akhir satu titik tetap (*fixed point*) atau *attractor*. Titik *attractor* inilah yang menyebabkan apapun jenis citra masukan maka hasil akhirnya akan sama, hanya posisi dan orientasi yang akan memengaruhi hasil akhir. Transformasi kontraktif dibutuhkan pada setiap iterasi untuk menghasilkan objek yang mirip dengan citra masukannya

2.4 Quadtree Decomposition

Konsep kerja IFS (*Iterated Fuction System*) adalah sekumpulan transformasi kontraktif yang memetakan bidang R^2 ke dalam dirinya sendiri (Putra, 2010). PIFS (*Partitioned Iterated Fuction System*) bertujuan untuk mencari kemiripan lokal daerah dengan cara mempartisi blok-blok jelajah dan ranah. Proses partisi akan terulang dilakukan secara berulang hingga ditemukan / tidak ditemukan kemiripan lokal. Kode fraktal dihasilkan pada transformasi fraktal tergantung dari metode partisi. Penelitian menggunakan metode partisi *quadtree*, karena metode ini mudah dan cukup populer untuk melakukan partisi secara rekursif. (Putra, 2010) Prinsip kerja *quadtree* adalah sebagai berikut citra biner berukuran $2^n \times 2^n$ yang terdiri dari piksel bernilai 1 (hitam) berupa objek dan nilai 0 (putih) berupa latar belakang. Pertama, keseluruhan citra sebagai node akar (*root node*). Jika *node* tidak terdiri 1 semuanya, atau 0 semuanya maka *node* tersebut disebut *gray node*, maka proses dekomposisi dilakukan. Proses dekomposisi akan membagi citra menjadi 4 bagian citra berukuran $2^{n-1} \times 2^{n-1}$. Setiap bagian citra akan diperiksa apakah perlu didekomposisi atau tidak. Bila setiap bagian semuanya bernilai 1 atau bernilai 0 maka proses dekomposisi tidak dilakukan, namun bila beragam maka dekomposisi akan dilakukan.

Quadtree pada penelitian ini dilakukan pada citra citra bukan biner, sehingga yang menentukan apakah suatu blok dipartisi atau tidak adalah nilai toleran ε , yang menentukan apakah blok tersebut mirip atau tidak. Bila blok ranah dan blok jelajah mirip maka tidak perlu dikomposisi, namun apabila tidak mirip akan dilakukan didekomposisi. Blok jelajah R adalah setengah dari blok ranah D. Untuk menentukan blok ranah mirip atau tidak dengan blok jelajah yaitu melalui persamaan (5), blok ranah yang terpilih adalah yang mempunyai jarak paling kecil dengan blok jelajah ditunjukkan pada persamaan (6). Selanjutnya jarak blok ranah dengan blok jelajah harus lebih kecil atau sama dengan nilai ambang atau nilai toleransi, seperti persamaan (7).

$$d(R, D) = \frac{1}{n} \sqrt{\sum_{i=1}^n (r_i - d_i)^2} \dots\dots\dots(5)$$

$$\min(d(R, D_i)), \text{ dengan } i = 1, \dots, k \dots\dots\dots(6)$$

$$d(R, D) \leq \varepsilon \dots\dots\dots(7)$$

2.5 Transformasi Kontraktif

Transformasi W dikatakan kontraktif apabila dua titik P1 dan P2 pada citra masukan menjadi lebih dekat pada citra hasil. Jarak keduanya didapatkan melalui persamaan (2). Jika transformasi tidak kontraktif maka titik-titik pada hasil akan menyebar keluar (tidak mengecil). Tranformasi kontraktif sering diawali dengan transformasi *affine*, untuk menghasilkan sekumpulan *attractor* (Putra, 2010). Transformasi *affine* menggunakan persamaan (3).

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \dots\dots\dots(2)$$

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} \dots\dots\dots(3)$$

w_i adalah operator transformasi x dan y adalah koordinat *pixel* pada citra, elemen $a_i, b_i, c_i, d_i, e_i, f_i$, merupakan koefisien transformasi *affine* untuk elemen perskalaan, pemutaran, kemiringan, dan kecondongan.

2.6 Huffman Encoding

Kode *huffman* adalah teknik untuk mengkompresi data dengan cara mengubah setiap karakter sebagai *string* biner secara optimal. *Huffman coding* adalah bentuk pengkodean statistik yang digunakan untuk mengurangi bit yang dibutuhkan untuk mewakili *string* simbol (Shweta Pandey, 2014). *Huffman encoding* dimulai dari membuat daftar semua simbol alfabet dalam urutan probabilitasnya dari urutan terkecil hingga terbesar kemunculannya (A.G.Ananth, 2012). Kemudian membangun sebuah pohon biner dengan simbol dari pada setiap daun dari atas ke bawah. Dimana disetiap langkah dua simbol dengan probabilitas terkecil digabungkan lalu ditambahkan ke bagian atas pohon parsial kemunculan dan diganti dengan simbol pembantu sebagai perwakilan dua simbol asli. Ketika daftar dikurangi menjadi hanya satu simbol dan mewakili seluruh alfabet, maka pohon itu selesai. Disetiap cabang ke daun diberi label “0” cabang ke kiri dan “1” cabang ke kanan.

2.7 Huffman Decoding

Pada tiap kode *huffman* yang dihasilkan adalah unik maka proses *decode* atau dekompresi dapat dilakukan dengan mudah dan sederhana. Misalnya dari runtun kode yang didapatkan, dibaca bit pertama yaitu 1. Kode ini tidak ada didalam sebuah tabel, maka dibaca kode berikutnya sampai terbentuk kode yang terdapat dalam tabel dan mendapatkan simbol. Proses ini dilakukan hingga semua bit dalam runtun kode *input* terbaca semua, sehingga didapatkan simbol yang terbentuk. Sebagai catatan untuk proses dekompresi tidak melibatkan pohon *huffman*, namun hanya membutuhkan runtun kode dan tabel konversi untuk masing-masing simbol (Dewanto, 2014).

2.8 Kualitas Pemampatan

Metode kompresi yang baik adalah metode yang mampu mengembalikan citra hasil kompresi menjadi citra semula tanpa kehilangan informasi apapun (T.Sutoyo, 2009). Suatu metode kompresi dapat dinilai dari kualitas citra hasil kompresi dan rasio kompresinya (Fazry, 2008). Matrik yang digunakan untuk mengukur perbedaan antara dua citra z^0 dan z' yang mempunyai masing-masing dimensi $m \times n$ adalah matrik *rms*, seperti yang ditunjukkan rumus persamaan (8).

$$rms = \frac{1}{MN} \sqrt{\sum_{i=1}^N \sum_{j=1}^M (z_{ij}^o - z_{ij}')^2} \dots\dots\dots(8)$$

Dimana, M dan N adalah dimensi masing-masing adalah tinggi dan lebar citra. z_{ij}^o dan z_{ij}' masing masing adalah nilai intensitas baris ke-i dan kolom ke-j dari citra hasil kompresi dan citra sebelum dikompresi. Kualitas citra dapat diukur secara kuantitatif menggunakan besaran PSNR (*Peak Signal to Noise Ratio*) (T.Sutoyo, 2009). PSNR merupakan perbandingan antara nilai maksimum dari sinyal dengan besar besarnya derau (Aditya Mahmud Faza, 2016). PSNR mempunyai satuan dB (*decibel*). Untuk mencari nilai PSNR dilakukan dengan rumus persamaan (9).

$$PSNR = 20 \cdot \log_{10} \left(\frac{maks}{rms} \right) \dots\dots\dots(9)$$

Dalam hal ini, nilai *maks* didapatkan dari nilai intensitas terbesar dan nilai *rms* didapatkan dari perhitungan dari persamaan (8). Semakin besar nilai PSNR maka kualitas citra hasil kompresi semakin baik (Fazry, 2008). Setelah didapatkan hasil validasi kualitas kompresi maka dapat dilakukan penghitungan rasio kompresi. Rasio kompresi citra adalah ukuran presentase citra yang telah berhasil dikompresi (T.Sutoyo, 2009). Rasio kompresi citra dapat dihitung dengan persamaan (10).

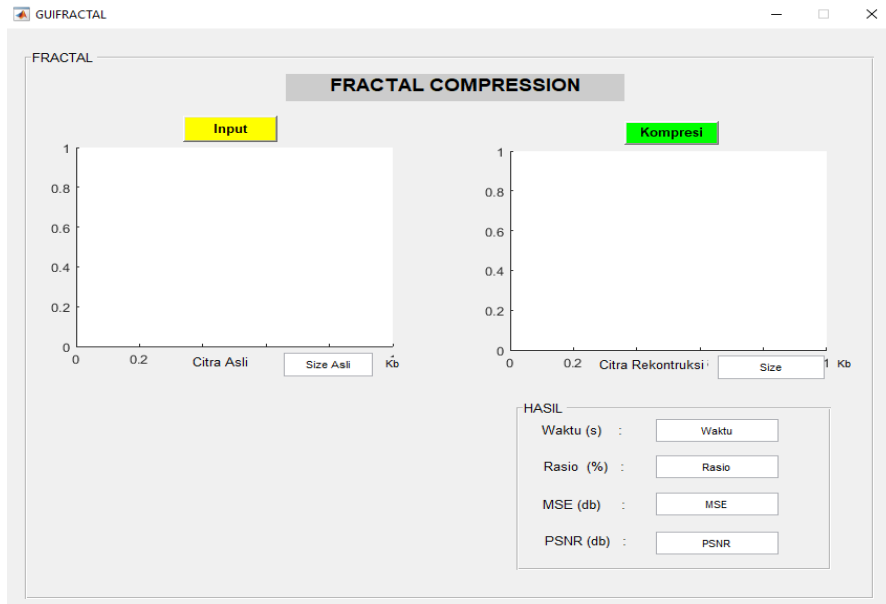
$$Rasio = 100\% - \left[\frac{ukuran\ hasil\ citra\ kompresi}{ukuran\ citra\ asli} \times 100\% \right] \dots\dots\dots(10)$$

Dengan kata lain ukuran citra menyatakan besar penggunaan memori untuk menyimpan citra. Semakin tinggi rasio kompresi yang dicapai maka metode kompresi semakin baik (Fazry, 2008).

3. HASIL DAN PEMBAHASAN

3.1 Tampilan Sistem GUI

Pada penelitian ini dibangun sebuah GUI yang dibuat dari aplikasi MATLAB 2016a yang dapat melakukan kompresi citra sesuai dengan metode yang digunakan. Objek citra yang digunakan sebagai data uji merupakan citra sekunder yang diperoleh dari internet seperti citra natural atau pemandangan alam sehingga didapatkan citra yang akan diproses menggunakan GUI yang sudah dibuat. Citra yang akan diolah merupakan citra yang mempunyai tingkat detail tinggi atau tingkat detail rendah yang akan dikompresi dengan menggunakan metode kompresi fraktal untuk mengetahui metode yang sesuai dengan kompresi citra pada saat ini. GUI Matlab dapat menampilkan ukuran citra sebelum dan sesudah dikompresi serta perhitungan nilai MSE dan PSNR untuk mengetahui kualitas citra yang didapatkan. GUI Matlab juga dapat menampilkan rasio kompresi dan menyimpan hasil dari kompresi citra tersebut secara otomatis. Tampilan aplikasi kompresi ditunjukkan pada Gambar 5.



Gambar 5. Tampilan Sistem GUI matlab

3.2 Cara kerja aplikasi dengan *Pseudocode*

Langkah kerja yang pertama adalah tahap *input* citra awal. Tahap *input* citra pertama adalah dengan memilih citra keabuan (*grayscale*) sebagai citra uji kemudian menampilkan citra dan ukuran citra awal, seperti yang ditunjukkan pada *pseudo-code* dibawah ini :

```

Start
    Ambil citra grayscale (I)
    Tampilkan pada frame sebagai citra asli.
  
```

Langkah kerja kedua adalah tahap kompresi. Tahap kompresi yang pertama adalah merubah ukuran citra menjadi 256 X 256 untuk mempermudah membagi blok *range*. Penggunaan citra keabuan (*grayscale*) dimaksudkan untuk mendapatkan *pixel* bernilai 1 (hitam) dan 0 (putih) untuk diproses pada *quadtree*. Setelah proses transformasi tahap selanjutnya adalah proses *quadtree decomposisi* untuk mempartisi apakah blok ranah dan jelajah mirip atau tidak, jika mirip maka tidak perlu didekomposisi, tetapi jika tidak mirip maka perlu didekomposisi, kemudian dilakukan proses fraktal yaitu *self-similarity*, berapapun skala/ukuran yang digunakan untuk menghasilkan objek fraktal maka objek yang dihasilkan akan tetap memiliki objek fraktal itu sendiri secara keseluruhan. Hasil iterasi *self-similarity* akan selalu lebih kecil daripada sebelumnya dan akan didapatkan hasil akhir satu titik tetap (*fixed point*). *Fixed point* inilah yang menyebabkan apapun jenis citra masukan maka hasil akhirnya akan tetap sama. Kemudian dilakukan langkah transformasi citra untuk elemen penyekalaan, kecondongan, pemutaran, dan penggeseran. Kemudian dilakukan proses dengan *huffman coding* untuk melakukan pendekatan statistik berdasarkan frekuensi kemunculan nilai *grayscale*, selanjutnya tahap kompresi dilakukan, seperti yang ditunjukkan *pseudo-code* dibawah ini :


```

%%quadtree decomposition
    penerapan dekomposisi quadtree pada citra_input(I), ambang = 1, dimensi minimum
    = 2, dimensi maksimum = 64
        s = qtdecomp(I, 0.1,[2 62])
        matriks untuk merekam nilai s
        [i, j, blksize] = find(s)
        blkcount = menghitung panjang i
        membuat matriks nol sesuai ukuran total blok
%%self-similarity untuk kode fractal
    for k=1:blkcount
        nilai mean2 (rata-rata) matriks range dengan domain.
    end for
    merubah avg tipe uint8
%%menyiapkan data
    pengaturan elemen matriks
    i(end+1) = 0;
    j(end+1) = 0;
    blksize(end+1)=0;
    matriks total (data)
    data dikonversi ke tipe single
    membuat variable baru (nilai acak) sesuai ukuran matriks data
    symbols = unique(data);
    mendapatkan frekuensi symbols yang ada pada variable data
    counts = hist(data, symbols);
    distribusi probabilitas (counts) dan skalanya
    membuat dictionary
%%huffman code compressing
    huffman code dictionary antara 'symbols' dengan distribusi probabilitas counts
    dict = huffmandict(symbols, p)
    dienkodkan kedalam huffman
    huffmanenco(data, dict)

```

Langkah ketiga merupakan tahap dekompresi. Tahap dekompresi adalah menyusun ulang hasil kompresi untuk ditampilkan. Pada tahap ini menggunakan *huffman decoding* untuk menyusun ulang hasil kompresi yang telah dilakukan sebelumnya dan menampilkan hasil dari dekompresi dan ukuran citra. Tahap dekompresi ditunjukkan *pseudo-code* dibawah ini :

```

%% huffman decoding
    decoding dari dictionary dengan encodingnya
    datanew = huffmandeco(comp,dict)
    variable data adalah nilai seluruh blok hasil dekomposisi
    zeroindex = find(data==0)
    nilai baris pada matriks datanew
        inew = datanew (1:zeroindex(1) -1)
    nilai kolom pada matriks datanew
        jnew = datanew(zeroindex(1) +1:zeroindex(2)-1)

```

```

nilai blocksize pada matriks datanew
    blksize = datanew(zeroindex(2) +1 : zeroindex(3)-1)
mendapatkan nilai rata-rata (mean)
    avgnew = datanew(zeroindex(3)+1:end);
%%rekonstruksi citra hasil kompresi
    variable avgnew diset bertipe uint8
    penggabungan nilai matriks i, j, dan blksize.

```

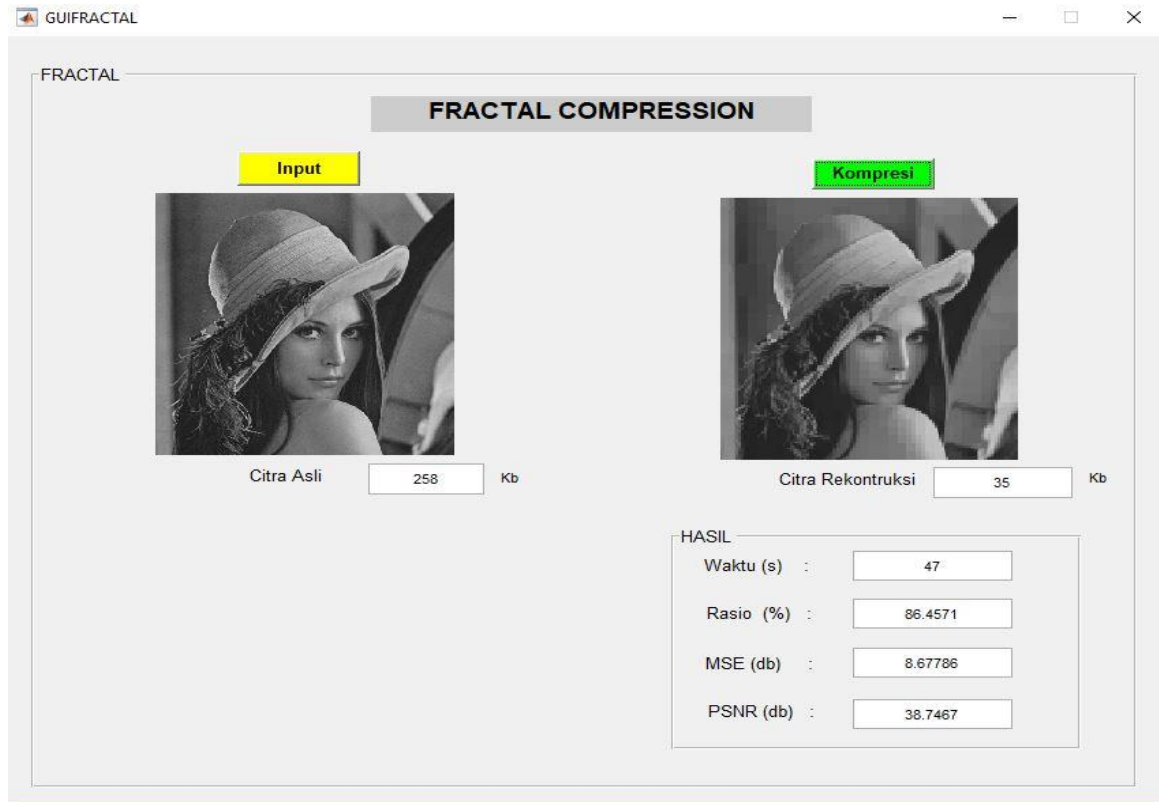
Langkah keempat adalah aplikasi menampilkan waktu dari lamanya proses kompresi sampai dekompresi. Aplikasi menampilkan rasio kompresi yang didapatkan dari hasil penghitungan selisih dari ukuran awal dan ukuran hasil kompresi, dan menampilkan PSNR dan MSE untuk menunjukkan kualitas citra, yang ditunjukkan *pseudo-code* dibawah ini :

```

%% Mean Square Error dan Peak Signal to Noise Ratio
menghitung nilai kualitas citra hasil kompresi
Citrakompresi (y)
MSE = sqrt(sum(sum(errorimg .* errorimg)) / (X * y));
PSNR = 10*log(255*255/MSE) / log(10);
%% ratio compression
perbandingan citra awal dengan citra hasil kompresi
citraHasil = length(comp)+8*length(symbols)+8*length(sp) .

```

Setelah dilakukan semua langkah kerja maka dihasilkan tampilan GUI dari hasil kompresi citra dengan metode fraktal seperti yang ditunjukkan pada Gambar 6.







Gambar 6. Menampilkan citra hasil kompresi dan semua hasil penghitungan

3.3 Hasil Pengujian

Setelah melakukan tahapan kompresi dilakukan maka didapatkan citra hasil kompresi sesuai dengan metode yang yang digunakan yaitu metode kompresi fraktal. Berikut adalah citra uji dan hasil kompresi citra fraktal dari berbagai jenis citra yang digunakan, seperti yang ditunjukkan pada Gambar 7 dan Gambar 8.

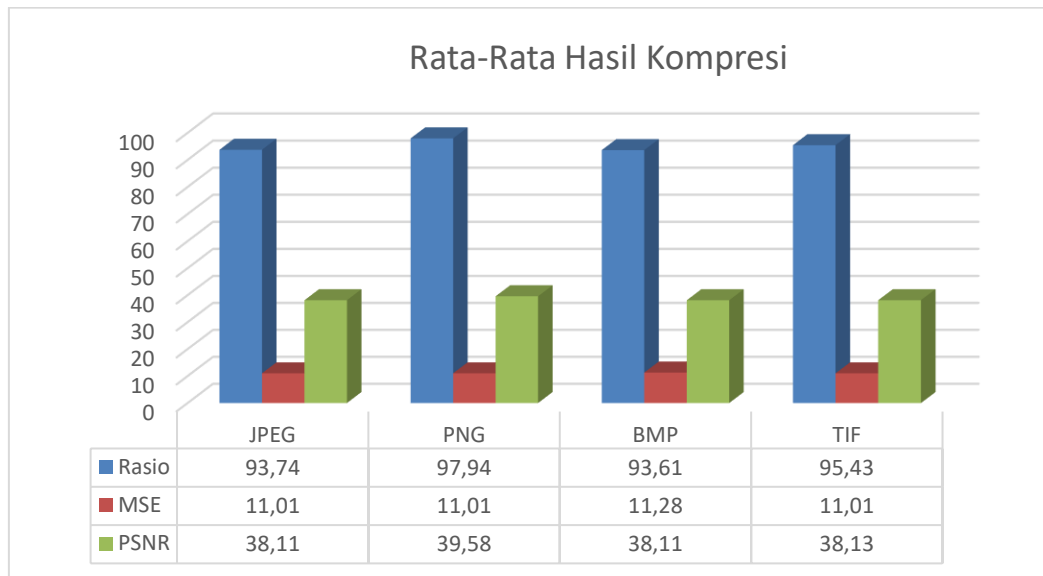


Gambar 7. Citra sebelum dikompresi

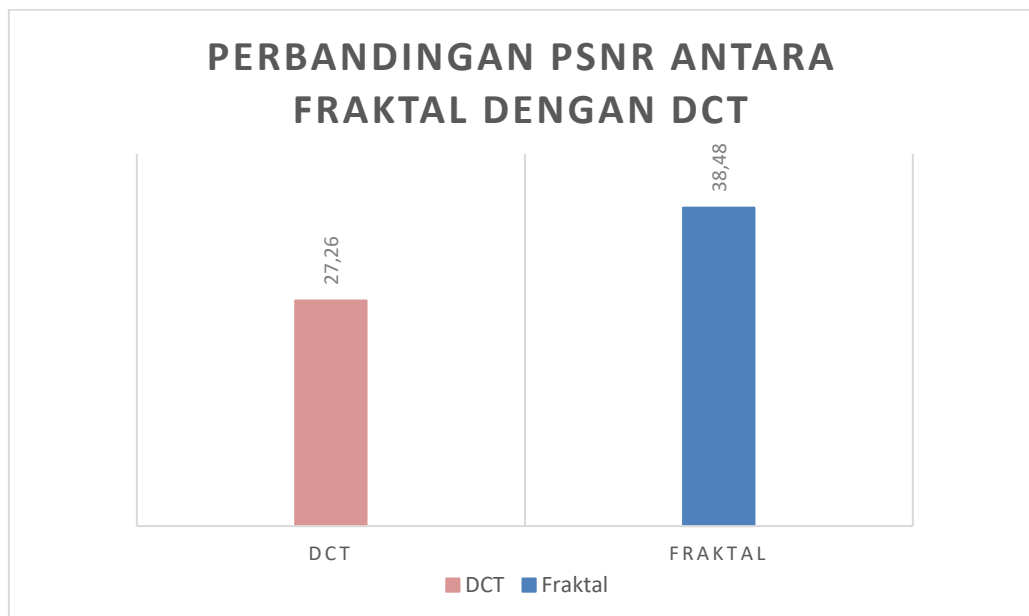
Lena (.jpg)	Lena (.bmp)	Lena (.tif)	Lena (.png)
			

Gambar 8. Citra hasil pengujian

Setelah citra uji terkompresi maka selanjutnya akan dilakukan penghitungan MSE dan PSNR. Penggunaan penghitungan MSE dan PSNR dalam penelitian ini adalah untuk mengetahui keberhasilan metode kompresi fraktal dengan keempat format gambar yang berbeda dan membandingkan metode fractal dengan metode kompresi lain seperti metode kompresi DCT. Berikut ini adalah hasil penghitungan nilai rata-rata citra uji dari metode kompresi citra fraktal dan dari metode kompresi DCT, seperti yang ditunjukan pada Gambar 9 dan Gambar 10.



Gambar 9. Rata-Rata Nilai MSE dan PSNR metode Fraktal pada jenis format citra yang berbeda



Gambar 10. Hasil perbandingan nilai PSNR

4. PENUTUP

4.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan maka dapat ditarik kesimpulan bahwa metode fraktal merupakan salah satu metode kompresi citra mampu mengurangi ukuran citra dengan kualitas citra dan rasio kompresi citra yang tinggi. Metode kompresi fraktal menghasilkan nilai rata-rata MSE adalah 11,07 dB dan nilai rata-rata PSNR adalah 38,48 dB serta rata-rata *ratio* kompresinya mencapai 95.05%. Hasil penelitian memperlihatkan citra output kompresi fraktal memiliki tingkat detail citra yang lebih rendah dibanding citra masukan.

4.2 SARAN

Dalam penelitian ini masih memiliki kelemahan yaitu waktu kompresi yang dihasilkan masih kurang bagus saat proses kompresi. Untuk penelitian selanjutnya bisa dikembangkan untuk mendapatkan waktu yang lebih baik seperti penggunaan kompresi RLE.

DAFTAR PUSTAKA

- A.G.Ananth, V. d. (2012). *Fractal Image Compression Using Quadtree Decomposition And Huffman Coding . International Journal (SIPIJ)*, 207-212.
- Aditya Mahmud Faza, C. S. (2016). Analisis Kinerja Kompresi Citra Digital dengan Komparasi DWT, DCT dan Hybrid (DWT-DCT) . 1-5.
- Dewanto, W. (2014). Teknik Kompresi Data. 1-85.
- Fazry, L. (2008). Kompresi Citra Fraktal Menggunakan Algoritma Genetika. *Skripsi*, 1-94.
- Harjoko, A. S. (2006). Kompresi Citra Digital Dengan Fraktal Sebagai Teknik Kompresi Alternatif . 1-6.
- Hiryanto, S. N. (2010). Algoritma Kompresi *Fraktal Sequential* Dan *Pararel* Untuk Kompresi Citra. *Jurnal Ilmu Komputer dan Informasi*, 208 -217.
- Mahmudi, A. (2017). Analisa Komparasi Teknik Kompresi Citra Medis Menggunakan *Run Lenght Encoding* (RLE), *Discrete Wavelet Transfrom* (DWT), *Discrete Cosine Transfrom* (DCT). *Tugas Akhir*, 1-17.
- Mukharrom Edisuryana, R. R. (2013). Aplikasi Steganografi Pada Citra Berformat *Bitmap* Dengan Menggunakan Metode *End Of File* . 735-742.
- Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: Andi.
- Shweta Pandey, M. S. (2014). *Hybrid Fractal Image Compression Using Quadtree Decomposition with Huffman Coding . International Journal of Science and Research (IJSR)* , 943-948.
- T.Sutoyo, S. M. (2009). Teori Pengolahan Citra Digital. Yogyakarta: Andi.
- Triana, D. (2015). Analisis Kompresi Citra Digital Dengan Metode Fraktal. *Skripsi*, 1-100.